Low degree representing polynomials of high sensitivity Boolean functions

Juris Smotrovs, Jurijs Zaicevs

University of Latvia
Faculty of Science and Technology

Estonian-Latvian Computer Science Theory Days, Tartu, April 26, 2026

Motivation

- Boolean functions with high sensitivity have high deterministic query complexity
- Boolean functions with low degree may have low quantum query complexity
- Thus functions with both of these properties are candidates for quantum advantage
- This topic may have/has other applications wherever Boolean function analysis is used (communication complexity, machine learning, etc.)

Some basic notions

- Boolean function: $f: \{0,1\}^n \rightarrow \{0,1\}$

$$f(x_1, x_2, \ldots, x_n) = f(x_1 x_2 \ldots x_n) = f(x)$$

- Hamming weight of an input tuple $x = (x_1, x_2, \ldots, x_n)$:

$$|x| \stackrel{\text{def}}{=} x_1 + x_2 + \cdots + x_n$$

Boolean functions with high sensitivity

- The sensitivity $s_x(f)$ of a Boolean function f on input $x = (x_1, x_2, \ldots, x_n)$: the number of such i that

$$f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) \neq f(x_1, \ldots, x_{i-1}, \neg x_i, x_{i+1}, \ldots, x_n)$$

- The sensitivity $s_x(f)$ of a Boolean function f:

$$s(f) = \max_{x \in \{0,1\}^n} s_x(f)$$

- Example:

$$OR_n(x_1, x_2, \ldots, x_n) = x_1 \vee x_2 \vee \cdots \vee x_n$$

$$s(OR_n) = n$$

because the input $00 \ldots 0$ is sensitive in all bits.

Boolean functions with high sensitivity

- The sensitivity $s_x(f)$ of a Boolean function f on input $x = (x_1, x_2, \ldots, x_n)$: the number of such i that

$$f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) \neq f(x_1, \ldots, x_{i-1}, \neg x_i, x_{i+1}, \ldots, x_n)$$

- The sensitivity $s_x(f)$ of a Boolean function f:

$$s(f) = \max_{x \in \{0,1\}^n} s_x(f)$$

- We will consider Boolean functions f with $s(f) = n$. To ensure that, WLOG we will suppose that

$$f(0^n) = 0 \text{ and } f(x) = 1 \text{ for all } x \text{ with } |x| = 1.$$

- (It is known that the deterministic query complexity of f is at least $s(f)$.)

Representing polynomials of Boolean functions

- The representing polynomial of a Boolean function: a real polynomial $p_f(x_1, x_2, \ldots, x_n)$ such that
$$\forall x \in \{0,1\}^n : p_f(x) = f(x)$$
- Can be assumed to be multilinear due to the identity $a^2 = a$ in the domain $\{0,1\}$
- There exists exactly one multilinear representing polynomial for every Boolean function
- Example. The representing polynomial of the «majority» function $\mathrm{MAJ}(x_1, x_2, x_3)$:
$$p_{MAJ}(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3 - 2x_1x_2x_3$$
- Degree of a Boolean function: $\deg(f) \stackrel{\text{def}}{=} \deg(p_f)$

Approximate polynomials of Boolean functions

- An approximate polynomial of a Boolean function: a real polynomial $\tilde{p}_f(x_1, x_2, \dots, x_n)$ such that

  $\forall x \in \{0,1\}^n$: $\left|\tilde{p}_f(x) - f(x)\right| \leq \varepsilon$ for some $0 < \varepsilon < 1/2$.

- Can be assumed to be multilinear due to the identity $a^2 = a$ in the domain $\{0,1\}$
- Is not necessarily unique
- Example. An approximate polynomial for the «majority» function $\mathrm{MAJ}(x_1, x_2, x_3)$ with $\varepsilon = 1/3$:

$$\tilde{p}_{MAJ}(x_1, x_2, x_3) = x_1 + x_2 + x_3$$

- Approximate degree of a Boolean function:

$$\widetilde{\deg}(f) \stackrel{\text{def}}{=} \min_{\tilde{p}_f \approx f} \deg(\tilde{p}_f)$$

Main problem of this research

- How low can the degree $d = \deg(f)$ be given that $s(f) = n$?
- Or, equivalently, how large can n be, given d?
- It is known that $\deg(f) \geq \sqrt[4]{\frac{3}{2}}\sqrt{n}$ (Nisan, Szegedy 1992, Tal 2013, Wellens, Ozols 2020)
- Similarly, how low can $\widetilde{\deg}(f)$ be?
- It is known that $\widetilde{\deg}(f) \geq \sqrt[4]{\frac{6}{101}}\sqrt{n}$ for $\varepsilon = {}^{1}/_{3}$ (Nisan, Szegedy 1992, Wellens 2020, Proskurin 2021)
- Find new functions with low $\deg(f)$ or $\widetilde{\deg}(f)$

Univariate polynomial R_f

- Symmetrization polynomial:

$$p_f^{sym}(x) = \frac{1}{n!} \sum_{\pi \in S_n} p_f(x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)})$$

- Minsky, Papert, 1969: There exists such univariate polynomial $R_f(t)$ that for all $x \in \{0,1\}^n$:

$$R_f(|x|) = p_f^{sym}(x)$$

- We have $\deg(R_f) = \deg(p_f^{sym}) \leq \deg(f)$
- I. Stepanovs in 2012 used the properties of R_f to prove with computer case analysis that there is no Boolean function with $n = s(f) = 15$ and $\deg(f) = 5$

Univariate polynomial Q_f

- Wellens, R. Ozols (independently), 2020:
$$Q_f(t) = p_f(t, t, \dots, t)$$
- We have $\deg(Q_f) = \deg(R_f) \leq \deg(f)$
- They used the properties of Q_f to prove that
$$\deg(f) \geq \sqrt[4]{3/2}\sqrt{n}$$

Univariate polynomial S_f

- Let

$$s_k = \sum_{x:\,|x|=k} f(x)$$

- s_k is an integer from $\left[0, \binom{n}{k}\right]$
- We introduce another univariate polynomial:

$$S_f(t) = \sum_{k=0}^{n} s_k t^k$$

- We prove that $d = \deg(Q_f)$ iff $(t+1)^{n-d}$ divides $S_f(t)$
- This lets us apply integer linear programming tools to look for possible polynomials S_f for given n, d

Numerical results

- Integer linear programming produced the following upper bounds on n, given d:

d	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
n	1	3	6	10	15	21	29	38	47	58	71	84	99	114	131	149	168

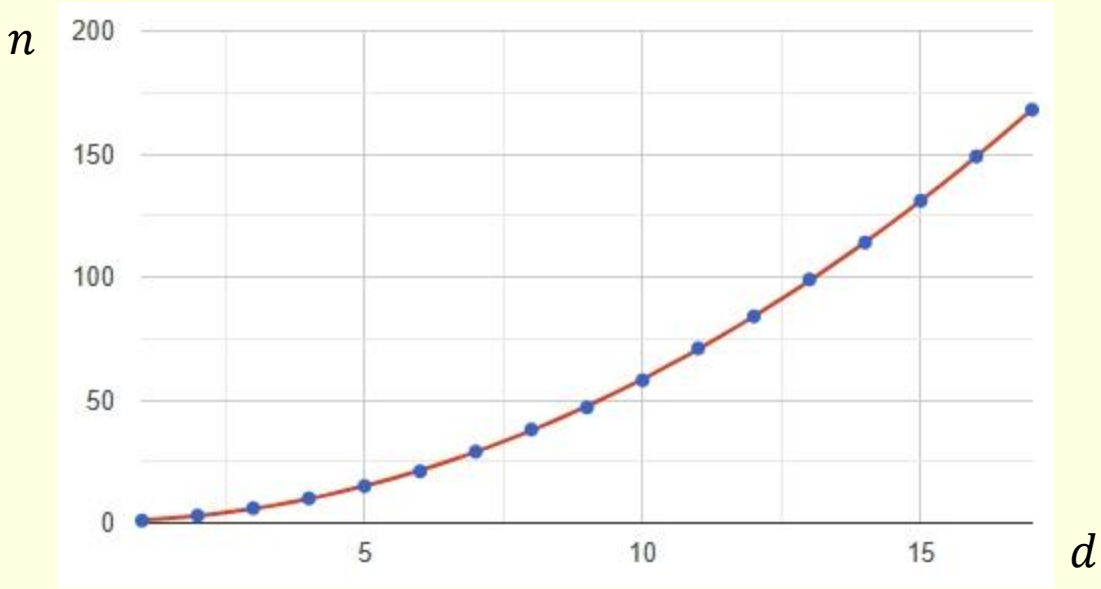
- Relaxation to rational solutions and solving the dual program allowed to verify these results as well as to extend the bounds interval up to $d = 33$:

d	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
n	189	210	233	257	282	308	336	364	394	425	457	490	524	560	596	634

Numerical results

- Fitting the integer linear program results produces the quadratic function

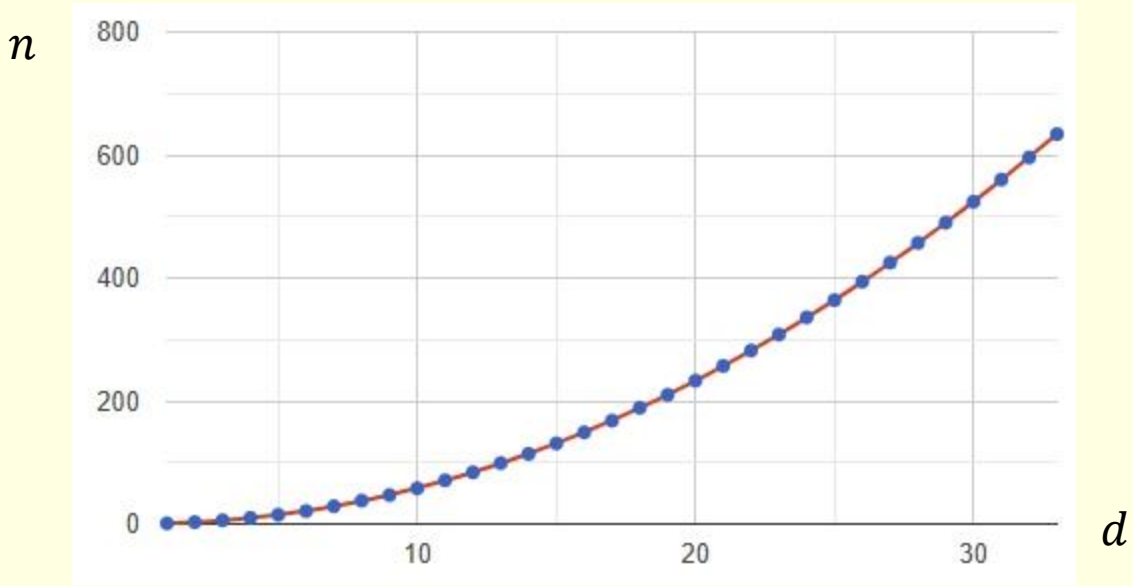
$$n = 0.6323529 - 0.02670279d + 0.5811404d^2$$



Numerical results

- Fitting the relaxed dual linear program results produces a very similar quadratic function

$$n = 0.629399 - 0.047756d + 0.583368d^2$$



Numerical results

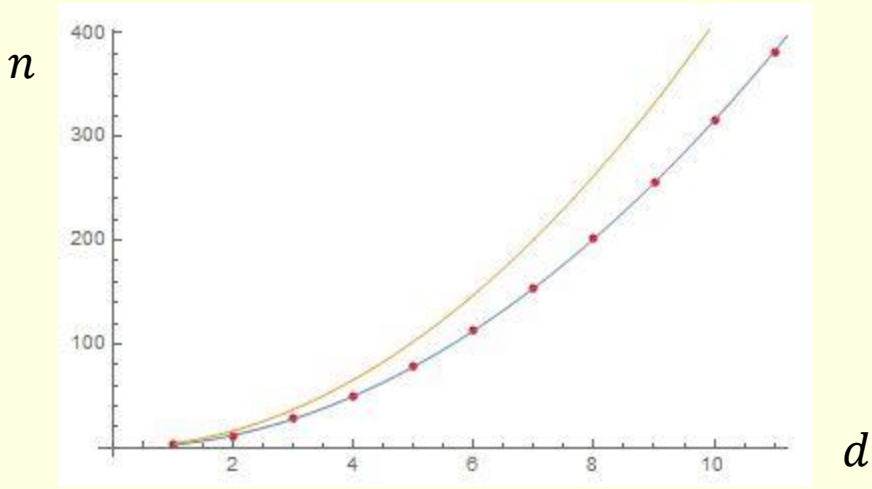
- These two functions are practically indistinguishable (violet line overlaying a practically invisible brownish line in the graph below). The red line is the best known theoretical bounds $n \leq \sqrt{2/3}\, d^2$:

Numerical results

- Similar analysis for approximate polynomials with $\varepsilon = {}^{1}/_{3}$ (up to $d = 11$) also produces bounds nicely fitting a quadratic function,

$$n = -0.121212 - 0.213287\, d + 3.18065\, d^2$$

Here it is together with theoretical bounds $n \leq \sqrt{{}^{101}/_{6}}\, d^2$ (the yellow line):



Looking for particular functions

- There is a Kushilevitz (1992) function with parameters

$$d = 3, n = 6$$

- Its iteration provides an infinite sequence of functions with

$$n = d^{\log 6/\log 3} = d^{1.6309\ldots}$$

- It is still the current record holder as a function sequence with the highest degree of d

Looking for particular functions

- Integer linear program provides potential statistics (by Hamming weight) of the Boolean functions
- E.g., if a function with $d = 5, n = 14$ exists (which would beat the Kushilevitz function), it must have one of 247 statistics (found already by I. Stepanovs)
- One of these statistics is rather remarkable, with symmetries and other interesting properties (e.g. all s_k except for s_0 and s_{13} are divisible by 13):

k=|x|	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
s_k	1	0	0	130	689	1638	2184	1716	819	364	312	234	91	14	0

- However, we did not succeed in constructing such function

Conclusion

- The obtained numerical results provide concrete upper bounds on n for values of d up to 33 that are better than the ones obtained from the best known general theoretical bounds.
- Numerical results nicely fit a quadratic curve, suggesting that either the actual dependency is indeed quadratic, or that other methods need to be employed to prove better bounds
- Numerical results suggest a better coefficient at d^2 than the one provided by theory both for exact (representing) and approximate polynomials

Future work

- The numerical solutions of linear programs could suggest an analytical solution that could improve the best known theoretical bounds
- Bounds for much larger values of d can be obtained by taking only a fraction of inequalities. Work already done shows that it introduces only a very small relaxation of the bounds
- The found statistics by Hamming weight could suggest how to build particular low degree high sensitivity Boolean functions.

Thank you for attention!

Questions?